First order logic (Ch. 8)

"Logic is invincible, because in order to combat logic it is necessary to use logic."

-- Pierre Boatroux
Announcements

Writing 3 due Wednesday

Projects talk!
Propositional logic builds sentences that relate various symbols with true or false.

Each symbol is simply a unique identifier, but you cannot “generalize” between them.

While this is fairly expressive, it is also quite cumbersome as each part of the environment might need many symbols associated with it.
Review: Propositional logic

For example: to express just the top left cell of this mindsweep, we would need to have:

\[
P1, 1, 1 \land \neg P1, 1, 2 \land \neg P1, 1, 3 \\
\land \neg P1, 1, 4 \land \neg P1, 1, 5 \land \neg P1, 1, 6 \\
\land \neg P1, 1, 7 \land \neg P1, 1, 8 \land \neg P1, 1, B
\]

Sadly in propositional logic we cannot relate these 9 symbols/literals together as “value of cell [1,1]” despite this relationship existing in the environment.
FO logic: definitions

Propositional logic has “propositions” that are either true or false.

First order logic has objects and the relation between them is what is important.

This can provide a more compact way of expressing the environment (but is more complicated as we are not restricted to T/F, so we cannot brute force search truth tables).
FO logic: definitions

There are two basic things in first order logic:

1. **Objects** which are some sort of noun or “thing” in the environment (e.g. teacher, bat)

2. **Relations** among objects, which can be:
   2.1. **Unary** (or properties) which relate to a single object (e.g. red, healthy, boring)
   2.2. **n-ary** which involve more than one
   2.3. **functions**, one “value” for an object
FO logic: definitions

We can represent any sentence with objects and relations, for example:

“I am sleepy today”
Object: I Relations: Sleepy, Today
Logic: Sleepy(Today(I))

“I howl at full moons”
Object: Me, Moon Relations: Full, Howl
Logic: Full(Moon) => Howl(Me)
Let's identify objects and relations in this: Walking My Fish
Objects:
Person, Car, Road, Fish, Leash

Relations:
Unary: Wet(Fish), Wet(Road), Wet(Car)
n-ary: OnTopOf(Person, Road), OnTopOf(Car, Road), OnTopOf(Fish, Road)
Functions: attached(Leash) = Fish
FO logic: definitions

You find objects and relations (what type):
FO logic: definitions

Objects:
StickPerson, Fish, Pole, Hat, SP'sLeftLeg?

Relations examples....
unary: Black(StickPerson)
n-ary: Hold(StickPerson, Fish),
      Hold(StickPerson, Pole)
functions:
    OnHead(StickPerson), LeftLeg(StickPerson)
FO logic: definitions

The “arguments” to relations might have an order relation.

For example: Hold(StickPerson, Fish) might imply “StickPerson holds Fish”.

This is not a symmetric relationship, so Hold(Fish, StickPerson) conveys a different meaning.
FO logic: definitions

We represent relations as a set of “tuples” (generalize “pair” for more than 2 elements)

For example the “Hold” relation might be:
\{<\text{StickPerson}, \text{Fish}>, <\text{StickPerson}, \text{Pole}>\}

For functions, we normally provide the result:
OnHead:
\(<\text{StickPerson}> \rightarrow \text{Hat}\)
\(<\text{Fish}> \rightarrow \text{String}\)
Objects and relations form the basis of first order logic, but we also expand our syntax with three things:

1. Quantifiers (existential and universal)
2. Variables (much in the math sense)
3. Equality (as in $=$ not $\equiv$)

Otherwise we have a similar syntax to propositional logic (implies, AND, OR, etc.)
Existential quantifier

The existential quantifier is $\exists$, which means “there exists ...”

For example, if I had a variable “$x$”, then...

$$\exists x \ Santa(x)$$

... means “Santa exists” or “Someone is Santa”

$$\exists x \ InClass(x) \land WantsALongerBreak(x)$$

... means “Someone in class wanted a longer break” or “At least one person in class ...”
Existential quantifier

A variable is a place-holder for any object

So if we had 3 objects, \{Sue, Alex, Devin\}, we could formally write:

$$\exists x \ Santa(x)$$

As...

$$Santa(Sue) \lor Santa(Alex) \lor Santa(Devin)$$

... or in English: “Someone is Santa”, “Santa is Sue, Alex or Devin”
The **universal quantifier** is denoted by $\forall$, and means “for all ...”

Thus, $\forall x \ Santa(x)$ ... means “Everyone is a Santa”

If our objects were again {Sue, Alex, Devin}, then this would mean:

$Santa(Sue) \land Santa(Alex) \land (Devin)$
Quantifiers

As $\exists$ is basically ORs and $\forall$ is ANDs, we can apply De Morgan's laws:

$$\neg \exists x \ Santa(x) \equiv \forall x \ \neg Santa(x)$$

In words “No Santa exists” is the same as “Everyone is not Santa” (or “No one is Santa”)

You can have multiple quantifiers as well:

$$\exists x \exists y \ Messaging(x, y) \equiv \exists x, y \ Messaging(x, y)$$

The above means “Some pair of people are messaging” or “There exists two people msg”
Quantifiers

The order of quantifiers also matters:
\( \forall x \exists y \ Mother(x) = y \) means “For every person \( x \), they have some mother \( y \)” or “All people have some mother”

However in the opposite order:
\( \exists y \forall x \ Mother(x) = y \) means “There is some person \( y \), who is the mother to everyone” or “Everyone has the same mother”
Quantifiers

Write these two sentences in logic:

1. “Someone is happy yet sleepy”

2. “Everyone in class is thinking”
Quantifiers

Write these two sentences in logic:

1. “Someone is happy yet sleepy”
   \[ \exists x \ Happy(x) \land Sleepy(x) \]

2. “Everyone in class is thinking”
   \[ \forall x \ InClass(x) \Rightarrow Thinking(x) \]

Normally this is the case:

For “\( \exists \)” you use \( \land \)
For “\( \forall \)” you use \( \Rightarrow \)
Equality

In logic, equality means two things are the same (much as it does in math)

For example, $Sue = Alex$ would imply Sue and Alex are the same people

This is often useful with variables:

$\forall x, y \neg (x = y) \Rightarrow \neg (\text{Midterm}(x) = \text{Midterm}(y))$

... which means “No two (different) people have the same midterm score”
Assumptions

Being completely expressive in first order logic can be difficult at times

In the last statement you need the “$\neg(x = y)$” to ensure that the variable does not reference the same person/object

However, in general two objects could be the same thing...
Assumptions

To formally express something as simple as: “My brothers are Bob and Jack” requires....

\[ \begin{align*}
\text{Brother}(James, Bob) \\
\land \text{Brother}(James, Jack) \\
\land \neg (Bob = Jack) \\
\land \forall x \text{ Brother}(James, x) \Rightarrow (x = Bob \lor x = Jack)
\end{align*} \]

This is overly complicated as we have to specify that everyone else is not my brother and that Jack and Bob are different people.
Assumptions

For this reason, we make 3 assumptions:
1. Objects are unique (i.e. \( \neg (Bob = Jack) \))

2. All un-said sentences are false
   - Thus, if I only say \( \text{Brother}(James, Bob) \)
     then I imply: \( \neg \text{Brother}(James, Jack) \)

3. Only objects I have specified exist
   (i.e. I assume a person \( Davis \) does not exist as I never mentioned them)
Assumptions

These assumptions make it easier to write sentences more compactly.

Under these assumptions, “My sisters are Alice and Grace” can be represented as:

\( \text{Sister}(James, Alice) \land \text{Sister}(James, Grace) \)

These assumptions do make it harder to say more general sentences, such as: “Two of my sisters are Alice and Grace”